# MA4832 Microprocessor Systems
# Lab Exercise – Analog-to-Digital Converter

**In this session, you will learn how to**

- Use Port E as an A/D converter
- Read analog values from Light dependent resistor (LDR) and Rotary potentiometer circuits
- Store converted digital readings into predefined addresses
- Switch between multi-channel and single-channel modes

1. Multi-channel mode

- Schematic of the sample circuit is shown in Fig. 1.
- Program: adc_two_ch.s

---

```
; adc_two_ch.s
; samples two ADC channels, PortE bit 4 (PE4) and bit 5 (PE5), using Sample Sequencer 2 (SS2)

; GPIO_PORTE and ADC0 address

GPIO_PORTE_AFSEL_R     EQU 0x40024420
GPIO_PORTE_DEN_R       EQU 0x4002451C
GPIO_PORTE_AMSEL_R     EQU 0x40024528

ADC0_ACTSS_R           EQU 0x40038000
ADC0_PC_R              EQU 0x40038FC4
ADC0_SSPRI_R           EQU 0x40038020
ADC0_EMUX_R            EQU 0x40038014
ADC0_SSMUX2_R          EQU 0x40038080
ADC0_SSCTL2_R          EQU 0x40038084
ADC0_IM_R              EQU 0x40038008
ADC0_PSSI_R            EQU 0x40038028
ADC0_RIS_R             EQU 0x40038004
ADC0_SSFIFO2_R         EQU 0x40038088
ADC0_ISC_R             EQU 0x4003800C

SYSCTL_RCGCGPIO_R      EQU 0x400FE608        ; GPIO run mode clock gating control

SYSCTL_RCGCADC_R       EQU 0x400FE638        ; ADC run mode clock gating control

               THUMB
               AREA    DATA, ALIGN=4
               EXPORT  Result [DATA,SIZE=4]
Result         SPACE   4


               AREA    |.text|, CODE, READONLY, ALIGN=2
               THUMB
               EXPORT  Start

Start

; initialize Port E
; activate clock for PortE

               LDR R1, =SYSCTL_RCGCGPIO_R    ; R1 = &SYSCTL_RCGCGPIO_R
               LDR R0, [R1]                  ;
```

```
                ORR R0, R0, #0x10            ; turn on GPIOE clock
                STR R0, [R1]
                NOP                          ; allow time for clock to finish
                NOP
                NOP

; no need to unlock PE4 and PE5

; select alternate function
                LDR R1, =GPIO_PORTE_AFSEL_R
                LDR R0, [R1]
                ORR R0, R0, #0x30            ; enable alternate function on PE4 and PE5
                STR R0, [R1]

; disable digital port
                LDR R1, =GPIO_PORTE_DEN_R
                LDR R0, [R1]
                BIC R0, R0, #0x30            ; disable digital I/O on PE4 and PE5
                STR R0, [R1]

; enable analog functionality
                LDR R1, =GPIO_PORTE_AMSEL_R
                LDR R0, [R1]
                ORR R0, R0, #0x30            ; enable PE4 analog function
                STR R0, [R1]

; activate clock for ADC0
                LDR R1, =SYSCTL_RCGCADC_R
                LDR R0, [R1]
                ORR R0, R0, #0x01            ; activate ADC0
                STR R0, [R1]

                BL Delay                     ; delay subroutine -> allow time for clock to finish

                LDR R1, =ADC0_PC_R
                LDR R0, [R1]
                BIC R0, R0, #0x0F            ; clear max sample rate field
                ORR R0, R0, #0x1             ; configure for 125K samples/sec
                STR R0, [R1]

                LDR R1, =ADC0_SSPRI_R
                LDR R0, =0x1023              ; SS2 is highest priority
                STR R0, [R1]

                LDR R1, =ADC0_ACTSS_R
                LDR R0, [R1]
                BIC R0, R0, #0x0004          ; disable SS2 before configuration to
                STR R0, [R1]                 ; prevent erroneous execution if a trigger event
                                             ; were to occur

                LDR R1, =ADC0_EMUX_R
                LDR R0, [R1]
                BIC R0, R0, #0x0F00          ; SS2 is software trigger
                STR R0, [R1]

                LDR R1, =ADC0_SSMUX2_R
                LDR R0, [R1]
                BIC R0, R0, #0x00FF          ; clear SS2 field
                ADD R0, R0, #0x89            ; set channel -> select input pin AIN8 and AIN9
                STR R0, [R1]

                LDR R1, =ADC0_SSCTL2_R
                LDR R0, =0x0060              ; configure sample -> not reading Temp sensor,
                                             ; not differentially sampled,
                STR R0, [R1]                 ; assert raw interrupt signal at the end of
                                             ; conversion, second sample is last sample

                LDR R1, =ADC0_IM_R
                LDR R0, [R1]
                BIC R0, R0, #0x0004          ; disable SS2 interrupts
                STR R0, [R1]
```

```
                LDR R1, =ADC0_ACTSS_R
                LDR R0, [R1]
                ORR R0, R0, #0x0004           ; enable SS2
                STR R0, [R1]


Loop
                LDR R1, =ADC0_PSSI_R
                MOV R0, #0x04                 ; initiate sampling in SS2
                STR R0, [R1]

                LDR R1, =ADC0_RIS_R           ; R1 = address of ADC Raw Interrupt Status
                LDR R0, [R1]                  ; check end of a conversion
                CMP     R0, #0x04             ; when a sample has completed conversion -> a raw
                                              ; interrupt is enabled

                BNE     Loop

                LDR R1, =ADC0_SSFIFO2_R       ; load SS2 result FIFO into R1
                LDR R2, =Result
                LDR R0,[R1]
                STR R0,[R2]                   ; store PE4 reading in addr 0x20000000 & 0x20000001

                LDR R0,[R1]
                ADD R2, R2, #04
                STR R0,[R2]                   ; store PE5 reading in addr 0x20000004 & 0x20000005

                LDR R1, =ADC0_ISC_R
                LDR R0, [R1]
                ORR R0, R0, #04               ; acknowledge conversion
                STR R0, [R1]

                B Loop

Delay                                         ; Delay subroutine
                MOV R7,#0x0F

Countdown
                SUBS R7, #1                   ; subtract and set the flags based on the result
                BNE Countdown

                BX LR                         ; return from subroutine

                ALIGN                         ; make sure the end of this section is aligned
                END                           ; end of file
```

---

- Load and run the program.
- Do the following steps:
  - (i) Set a multimeter to measure DC voltage and use it to read the analog voltages on PE4 and PE5 input pins.
    (PE4 is connected to a LDR and PE5 is connected to a rotary potentiometer)
  - (ii) Record the readings obtained and do analog to digital conversions
  - (iii) Compare the calculated digital values to the digital readings stored at the following addresses:

    LDR sensor values – at 0x20000000 (low byte) & 0x20000001(high byte)
    Rotary potentiometer – at 0x20000004 (low byte) & 0x20000005 (high byte)

  - (iv) Change the intensity of light falling on LDR or turn the rotary potentiometer and watch the digital values change in these addresses.

2. Single Channel Mode

- Schematic of the sample circuit is the same as Fig. 1.
- Program: adc_single_ch.s

---

```
; adc_single_ch.s
; samples one ADC channel, PortE bit 4 (PE4), using Sample Sequencer 3(SS3)

; GPIO_PORTE and ADC0 address

GPIO_PORTE_AFSEL_R     EQU 0x40024420
GPIO_PORTE_DEN_R       EQU 0x4002451C
GPIO_PORTE_AMSEL_R     EQU 0x40024528

ADC0_ACTSS_R           EQU 0x40038000
ADC0_PC_R              EQU 0x40038FC4
ADC0_SSPRI_R           EQU 0x40038020
ADC0_EMUX_R            EQU 0x40038014
ADC0_SSMUX3_R          EQU 0x400380A0
ADC0_SSCTL3_R          EQU 0x400380A4
ADC0_IM_R              EQU 0x40038008
ADC0_PSSI_R            EQU 0x40038028
ADC0_RIS_R             EQU 0x40038004
ADC0_SSFIFO3_R         EQU 0x400380A8
ADC0_ISC_R             EQU 0x4003800C

SYSCTL_RCGCGPIO_R      EQU 0x400FE608        ; GPIO run mode clock gating control

SYSCTL_RCGCADC_R       EQU 0x400FE638        ; ADC run mode clock gating control

               THUMB
               AREA    DATA, ALIGN=4
               EXPORT  Result [DATA,SIZE=4]
Result         SPACE   4


               AREA    |.text|, CODE, READONLY, ALIGN=2
               THUMB
               EXPORT  Start

Start
               ; initialize Port E
               ; activate clock for PortE

               LDR R1, =SYSCTL_RCGCGPIO_R    ; R1 = &SYSCTL_RCGCGPIO_R
               LDR R0, [R1]                  ;
               ORR R0, R0, #0x10             ; turn on GPIOE clock
               STR R0, [R1]
               NOP                           ; allow time for clock to finish
               NOP
               NOP

               ; no need to unlock PE4

               ; enable alternate function
               LDR R1, =GPIO_PORTE_AFSEL_R
               LDR R0, [R1]
               ORR R0, R0, #0x10             ; enable alternate function on PE4
               STR R0, [R1]

               ; disable digital port
               LDR R1, =GPIO_PORTE_DEN_R
               LDR R0, [R1]
               BIC R0, R0, #0x10              ; disable digital I/O on PE4
               STR R0, [R1]
```

```
                ; enable analog function
                LDR R1, =GPIO_PORTE_AMSEL_R
                LDR R0, [R1]
                ORR R0, R0, #0x10              ; enable PE4 analog function
                STR R0, [R1]

                ; activate clock for ADC0
                LDR R1, =SYSCTL_RCGCADC_R
                LDR R0, [R1]
                ORR R0, R0, #0x01               ; activate ADC0
                STR R0, [R1]

                BL Delay                       ; delay subroutine -> allow time for clock to finish

                LDR R1, =ADC0_PC_R
                LDR R0, [R1]
                BIC R0, R0, #0x0F              ; clear max sample rate field
                ORR R0, R0, #0x1               ; configure for 125K samples/sec
                STR R0, [R1]

                LDR R1, =ADC0_SSPRI_R
                LDR R0, =0x0123                ; SS3 is highest priority
                STR R0, [R1]

                LDR R1, =ADC0_ACTSS_R
                LDR R0, [R1]
                BIC R0, R0, #0x08              ; disable SS3 before configuration to
                STR R0, [R1]                   ; prevent erroneous execution if a trigger event
were to occur

                LDR R1, =ADC0_EMUX_R
                LDR R0, [R1]
                BIC R0, R0, #0xF000            ; SS3 is software trigger
                STR R0, [R1]

                LDR R1, =ADC0_SSMUX3_R
                LDR R0, [R1]
                BIC R0, R0, #0x000F            ; clear SS3 field
                ADD R0, R0, #9                 ; set channel -> select input pin AIN9
                STR R0, [R1]

                LDR R1, =ADC0_SSCTL3_R
                LDR R0, =0x0006                ; configure 1st sample -> not reading Temp sensor,
                                               ; not differentially sampled,
                STR R0, [R1]                   ; assert raw interrupt signal at the end of
                                               ; conversion, first sample is last sample

                LDR R1, =ADC0_IM_R
                LDR R0, [R1]
                BIC R0, R0, #0x0008            ; disable SS3 interrupts
                STR R0, [R1]

                LDR R1, =ADC0_ACTSS_R
                LDR R0, [R1]
                ORR R0, R0, #0x0008            ; enable SS3
                STR R0, [R1]

Loop
                LDR R1, =ADC0_PSSI_R
                MOV R0, #0x08                  ; initiate sampling in SS3
                STR R0, [R1]

                LDR R1, =ADC0_RIS_R            ; R1 = address of ADC Raw Interrupt Status
                LDR R0, [R1]                   ; check end of a conversion
                CMP R0, #0x08                  ; when a sample has completed conversion -> a raw
                                               ; interrupt is enabled
                BNE Loop

                LDR R1, =ADC0_SSFIFO3_R        ; load SS3 result FIFO into R1
                LDR R0,[R1]
```

```
                LDR R2, =Result
                STR R0,[R2]

                LDR R1, =ADC0_ISC_R
                LDR R0, [R1]
                ORR R0, R0, #08              ; acknowledge conversion
                STR R0, [R1]

                B Loop

Delay                                        ; Delay subroutine
                MOV R7,#0x0F

Countdown
                SUBS R7, #1                  ; subtract and set the flags based on the result
                BNE Countdown

                BX LR                        ; return from subroutine

                ALIGN                        ; make sure the end of this section is aligned
                END                          ; end of file
```

---

- Load and run the program
- Observe the following points:
  (i) Only the LDR sensor reading has been converted.
  (ii) The sampling control and data capture is handled by the Sample Sequencer 3 (SS3).

**Exercise 1:**

For sample program 2, “adc_single_ch.s”, replace the LDR sensor with the potentiometer (pot), i.e. now you are reading from PE5, not PE4.

Modify the sample program 2 such that

- When the pot output rises beyond 1.0V, the on-board green LED (PortF 3) would start to flicker. When the pot output voltage rises beyond 2.0V, the red LED (PortF 1) would start to flicker (Turn off Green LED). When the voltage falls back to below 2.0V, the red LED is turned off. Falling below 1.0V would turn off the green LED too.

**Exercise 2:**

Modify the Exercise 1 program to sample and capture data from PE5 using Sample Sequencer 1 (SS1)

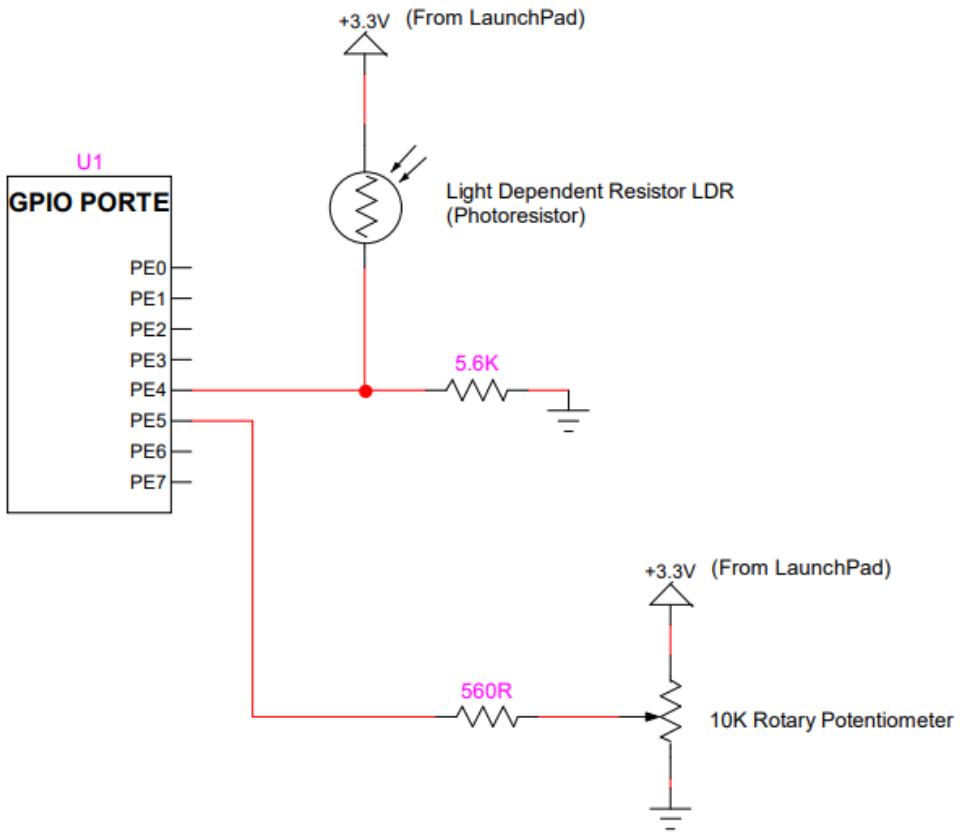


Fig. 1 Schematic Diagram